

Sistem Pengenalan Kata dengan Menggunakan *Linear Predictive Coding* dan *Nearest Neighbor Classifier*

Thiang, Hadi Saputra

Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra
Email: thiang@petra.ac.id

Abstrak

Makalah ini memaparkan tentang pengembangan sistem pengenalan kata dengan menggunakan cara pendekatan pola spektrum suara yaitu pencocokan pola suara manusia yang telah dibuat sebelumnya dengan pola suara yang baru masuk. Pengenalan suara dalam sistem ini dibatasi hanya suara untuk satu kata saja dan hanya lima kata yang akan dikenali yaitu kata “satu”, “dua”, “tiga”, “empat” dan “lima”. Input sistem diperoleh dari sebuah mikrofon. Output mikrofon diperkuat dengan menggunakan sebuah rangkaian amplifier yang kemudian diubah menjadi data digital dengan menggunakan analog to digital converter (ADC) untuk dapat dibaca oleh mikroprosesor TMS320C5x. Metode ekstraksi pola suara yang digunakan adalah linear predictive coding (LPC). Proses pengenalan kata dilakukan dengan menggunakan metode nearest neighbor classifier. Prinsip utama metode ini adalah mencari jarak terdekat antara pola suara yang akan dikenali dengan pola suara yang ada dalam database. Keputusan diambil dari pola suara dalam database yang memiliki jarak paling dekat dengan pola suara yang akan dikenali. Beberapa pengujian telah dilakukan antara lain pengujian sistem dengan suara orang yang ada dalam database, pengujian dengan suara orang yang tidak ada dalam database. Hasil pengujian menunjukkan bahwa sistem yang telah didisain dapat mengenali kata lebih dari 92%.

Kata Kunci: pengenalan kata, linear predictive coding, nearest neighbor classifier, TMS320C5X

Abstract

This paper describe about the development of word recognition system by using a voice matching approach, which will match between reference voice spectrum and unknown voice spectrum. This designed system is limited to recognize a voice, which only consists of one word. There are five words, which will be recognized in this system. They are “satu”, “dua”, “tiga”, “empat”, and “lima”. Input device of the system is a microphone. The output of microphone will be amplified by using a pre-amplifier circuit. Then, the signal is converted to digital signal by using an analog to digital converter (ADC), so that, the microprocessor TMS320C5x can read the data from the signal. This system uses linear predictive coding (LPC) to extract the pattern of the voice. Word recognition process is done by using nearest neighbor classifier method. Basic principle of this method is to find the shortest distance between unknown voice pattern and reference voice patterns. An unknown voice pattern is recognized as the reference voice pattern, which has the shortest distance with that unknown pattern. Some experiments have been done in various people voices. They are people whose voices are in reference voice database, people whose voices are not in database. Experimental results show that the system can recognize the word up to 92 %

Keywords: word recognition, linear predictive coding, nearest neighbor classifier, TMS320C5x

Pendahuluan

Dengan adanya kemajuan teknologi yang begitu pesat, saat ini untuk menjalankan suatu aplikasi cukup dengan mengucapkan sebuah kata saja contohnya seperti pada aplikasi *voice dialing*. Sebenarnya hal ini dapat dilakukan dengan sebuah komputer. Namun untuk membeli sebuah komputer adalah suatu hal yang tidak praktis, ukurannya relatif besar dan mahal harganya. Karena itu dalam makalah ini dipaparkan pengembangan suatu sistem

pengenalan kata dengan menggunakan sebuah mikroprosesor dengan harga yang lebih murah dan ukuran relatif kecil.

Metode yang digunakan dalam sistem ini adalah metode *linear predictive coding* (LPC) dan *nearest neighbor classifier*. LPC digunakan untuk mengekstrak pola kata dari suara yang di-input-kan. Sedangkan metode *nearest neighbor classifier* digunakan untuk mengenali kata yang di-input-kan. Sistem yang telah didisain diimplementasikan pada mikroprosesor TMS320C5x yang merupakan proses khusus untuk pemrosesan sinyal *digital*.

Selanjutnya pada bagian kedua makalah ini akan menjelaskan tentang sistem yang telah didisain mulai

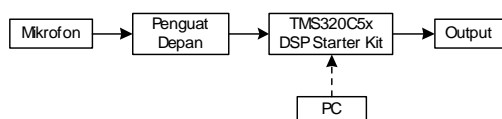
Catatan: Diskusi untuk makalah ini diterima sebelum tanggal 1 Juni 2005. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 5, nomor 2, September 2005.

dari perangkat keras, perangkat lunak dan sistem secara umum. Bagian ketiga dari makalah ini akan menjelaskan hasil pengujian yang telah dilakukan untuk sistem yang telah dibuat. Dan makalah ini akan ditutup dengan diskusi dan kesimpulan yang telah diambil berdasarkan hasil pengujian yang telah dilakukan.

Sistem Pengenalan Kata

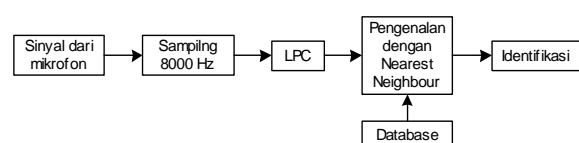
Pada dasarnya konsep perencanaan sistem pengenalan kata dapat dilihat pada blok diagram perangkat keras yang ditunjukkan oleh gambar 1 dan blok diagram sistem pengenalan kata yang ditunjukkan oleh gambar 2.

Secara garis besar, cara kerja perangkat keras adalah input suara manusia yang masuk akan diterima oleh mikrofon dengan tujuan untuk mengubah sinyal suara menjadi sinyal listrik. Karena sinyal output dari mikrofon sangat lemah, maka diperlukan sebuah rangkaian yang berfungsi untuk memperkuat sinyal dari mikrofon tersebut. Rangkaian tersebut disebut sebagai rangkaian penguat depan.



Gambar 1. Blok Diagram Perangkat Keras Sistem Pengenalan Kata

Kemudian sinyal dari mikrofon yang telah diperkuat oleh rangkaian *pre-amplifier* akan di-*sampling* oleh *analog to digital converter* (ADC) yang berfungsi mengubah sinyal listrik analog menjadi sinyal digital. Rangkaian ADC ini sudah terdapat di dalam TMS320C5x DSP *starter kit*. Sinyal digital dari ADC ini akan diolah untuk dikenali dan kemudian hasilnya ditampilkan di LCD.



Gambar 2. Blok Diagram Sistem Pengenalan Kata

Sinyal analog dari mikrofon di-*sampling* dengan frekuensi 8000 Hz. Pemilihan frekuensi *sampling* sebesar 8000 Hz karena frekuensi terbesar dari sinyal yang akan di-*sampling* adalah 3400 Hz sesuai dengan frekuensi suara manusia sekitar 600 – 3400 Hz. Hal ini dikarenakan adanya syarat bahwa frekuensi *sampling* minimal dua

kali dari frekuensi sinyal yang akan di-*sampling*. Lama waktu *sampling* data adalah setengah detik dengan demikian dalam satu kali *sampling* akan diperoleh 4000 data. Kemudian data ini akan diproses dengan analisa LPC sehingga menghasilkan koefisien-koefisien *cepstral* yang menjadi parameter untuk dikenali. Metode yang digunakan untuk pengenalan kata adalah metode *nearest neighbor classifier*.

Linear Predictive Coding (LPC)

Ide dasar dari LPC adalah bahwa sebuah sinyal suara manusia pada waktu n , $s(n)$, dapat diperkirakan sebagai kombinasi linier dari p sinyal suara manusia sebelumnya. Hal ini direpresentasikan dengan menggunakan persamaan berikut[1]:

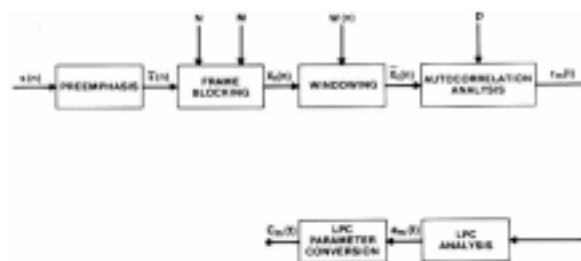
$$s(n) \approx a_1 \cdot s(n-1) + a_2 \cdot s(n-2) + \dots + a_p \cdot s(n-p) \quad (1)$$

dimana koefisien-koefisien a_1, a_2, \dots, a_p merupakan asumsi konstanta dari sebuah bagian dari analisa suara manusia. Persamaan 1 dapat diubah dengan menambahkan sebuah eksitasi $G \cdot u(n)$ sehingga menjadi:

$$s(n) = \sum_{i=1}^p a_i \cdot s(n-i) + G \cdot u(n) \quad (2)$$

dimana $u(n)$ adalah eksitasi yang dinormalisasi dan G adalah penguatan dari eksitasi tersebut. Jika persamaan 2 ini diekspresikan dalam domain z maka persamaan tersebut berubah menjadi:

$$S(z) = \sum_{i=1}^p a_i \cdot z^{-i} \cdot S(z) + G \cdot U(z) \quad (3)$$



Gambar 3. Blok Diagram Prosesor LPC

Gambar 3 menunjukkan blok diagram prosesor LPC yang biasa digunakan dalam sistem pengenalan suara manusia. Langkah-langkah dasar yang harus dilakukan mengenai blok diagram tersebut adalah:

a. Preemphasis

Sinyal suara yang telah diubah menjadi sinyal digital, $s(n)$, dilewatkan pada sebuah filter yang berorde rendah. Rangkaian *preemphasis* yang

paling sering digunakan adalah sebuah sistem orde satu yaitu:

$$H(z) = 1 - \tilde{a}.z^{-1} \quad 0.9 \leq a \leq 1 \quad (4)$$

Dengan demikian output dari rangkaian filter tersebut, $\tilde{s}(n)$, dalam fungsi input, $s(n)$, adalah sebagai berikut:

$$\tilde{s}(n) = s(n) - \tilde{a}.s(n-1) \quad (5)$$

b. *Frame Blocking*

Pada tahap ini sinyal yang telah di-*preemphasis*, diblok menjadi beberapa bagian dengan jumlah sampel N , dan tiap bagian dipisahkan dengan sejumlah M sampel. Gambar 4 menunjukkan penggambaran sinyal yang telah diblok dengan nilai $M=(1/3)N$. Bagian pertama terdiri dari sejumlah N sampel, kemudian bagian kedua dimulai dari sampel ke M juga sejumlah N sampel dan begitu seterusnya. Dengan demikian ada sinyal yang *overlap* dari setiap bagian sinyalnya. Hal ini memberikan hasil spektrum LPC yang akan berkorelasi tiap bagiannya.

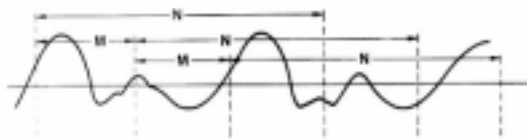
c. *Windowing*

Langkah berikutnya adalah melakukan proses *window* pada setiap bagian sinyal yang telah dibuat sebelumnya. Hal ini dilakukan untuk meminimalkan diskontinuitas pada bagian awal dan akhir sinyal. Jika didefinisikan sebuah *window* $w(n)$ dan sinyal tiap bagian adalah $x(n)$ maka sinyal hasil proses *windowing* ini adalah sebagai berikut:

$$\tilde{x}(n) = x(n).w(n) \quad 0 \leq n \leq N-1 \quad (6)$$

Model *window* yang paling sering digunakan untuk model LPC dengan metode autokorelasi adalah *Hamming Window* yang direpresentasikan dengan persamaan berikut:

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{N-1}\right) \quad (7)$$



Gambar 4. *Frame Blocking* sinyal

d. *Analisa Autokorelasi*

Tiap bagian yang telah diberi *window* kemudian akan dibentuk autokorelasinya dengan menggunakan persamaan berikut:

$$r(m) = \sum_{n=0}^{N-1-m} \tilde{x}(n) \cdot \tilde{x}(n+m)$$

$$m = 0, 1, 2, \dots, p \quad (8)$$

dimana nilai tertinggi dari autokorelasi tersebut, p , adalah orde dari analisa LPC yang akan dilakukan. Nilai yang umum untuk orde analisa LPC adalah antara 8 sampai 16. Keuntungan dari penggunaan metode autokorelasi adalah bahwa nilai ke-nol, $r(0)$, adalah energi dari sinyal yang dibuat autokorelasinya.

e. *Analisa LPC*

Langkah berikutnya adalah analisa LPC dimana semua nilai autokorelasi yang telah dihitung pada tahap sebelumnya akan diubah menjadi sebuah parameter LPC. Parameter ini bermacam-macam, ada disebut dengan nama koefisien LPC, koefisien pantulan (PARCOR), koefisien *cepstral*, atau transformasi lain yang diinginkan. Metode yang umum untuk menyelesaikan koefisien autokorelasi di atas menjadi koefisien LPC adalah dengan menggunakan metode Durbin yang algoritmanya adalah sebagai berikut:

$$E^{(0)} = r(0) \quad (9)$$

$$k_i = \frac{\left\{ r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} \cdot r(i-j) \right\}}{E^{(i-1)}} \quad 1 \leq i \leq p \quad (10)$$

$$\alpha_i^{(i)} = k_i \quad (11)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \cdot \alpha_{i-j}^{(i-1)} \quad (12)$$

$$E^{(i)} = (1 - k_i^2) \cdot E^{(i-1)} \quad (13)$$

Kemudian koefisien LPC yang dihasilkan adalah sebagai berikut:

$$a_m = \alpha_m^{(p)} \quad 1 \leq m \leq p \quad (14)$$

f. *Pengubahan parameter LPC menjadi koefisien cepstral*

Parameter LPC yang sangat penting yang bisa diturunkan dari koefisien LPC adalah koefisien *cepstral* LPC, $c(m)$. Persamaan yang digunakan untuk menghitungnya adalah:

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m} \right) \cdot c_k \cdot a_{m-k} \quad 1 \leq m \leq p \quad (15)$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m} \right) \cdot c_k \cdot a_{m-k} \quad m > p \quad (16)$$

Koefisien *cepstral* ini adalah koefisien dari representasi transformasi Fourier pada spektrum logaritmis.

Perhitungan-perhitungan LPC di atas menggunakan pemilihan $N=240$, $M=80$, $p=10$, dan $Q=12$ [1].

Nearest Neighbor Classifier

Prinsip dasar dari metode nearest neighbor adalah mengukur jarak antara pola sinyal yang tidak diketahui dengan pola sinyal referensi yang ada dalam database. Untuk menghitung jarak tersebut digunakan persamaan jarak *Euclidean* yang merupakan salah satu implementasi dalam algoritma *nearest neighbor*. Persamaan jarak *Euclidean* direpresentasikan dengan menggunakan persamaan berikut:

$$d_i(x, y) = \sum_i (x_i - y_i)^2 \quad (17)$$

Dimana x_i adalah pola sinyal yang tidak diketahui, y_i adalah pola sinyal referensi dalam database yang telah dibuat polanya terlebih dahulu, dan d_i adalah jarak antara sinyal yang tidak diketahui dengan pola sinyal referensi yang telah disimpan sebelumnya. Kemudian setelah didapat jarak antara pola sinyal yang tidak diketahui dengan pola sinyal referensi maka pengambilan keputusan ditentukan dengan menggunakan persamaan:

$$i = \min d_i \quad (18)$$

Nilai i adalah indeks dari pola sinyal referensi yang mempunyai jarak terdekat dengan pola sinyal yang tidak diketahui (sinyal *input*). Dengan demikian sinyal input akan dikenali sebagai pola sinyal referensi dengan indeks i .

Hasil Pengujian Sistem

Beberapa pengujian sistem telah dilakukan antara lain pengujian sistem untuk pengucapan kata secara berulang-ulang, pengujian sistem untuk pengucapan kata secara acak dan terakhir pengujian sistem untuk sumber yang berbeda dengan pola sinyal referensi dalam database. Dalam semua pengujian ini dibatasi hanya untuk lima kata yaitu kata “satu”, “dua”, “tiga”, “empat” dan “lima”. Pengujian sistem dilakukan dengan kondisi lingkungan sekitar yang cukup sunyi sehingga dapat dianggap tidak ada *noise* dari luar. Pembuatan pola sinyal referensi dilakukan dengan sederhana yaitu dengan memberi input kata lewat mikrofon kemudian disimpan dalam bentuk *file* teks dengan bantuan dari *debugger* TMS320C5x DSK. Kemudian *file* ini dimasukkan dalam program *assembly* yang telah dibuat.

Pengujian Sistem dengan Pengucapan Kata secara Berulang-ulang

Pada pengujian ini, pengucapan kata dilakukan secara berulang-ulang sebanyak 10 kali. Setelah selesai pengucapan untuk satu kata sebanyak 10 kali, dilanjutkan dengan pengucapan kata berikut berulang-ulang sebanyak 10 kali juga. Dalam pengujian ini, kata-kata yang diuji, diucapkan oleh orang yang sama dengan orang yang mengucapkan kata sebagai referensi dalam database. Berikut tabel 1 sampai tabel 5 menunjukkan hasil pengujian sistem yang telah dilakukan.

Tabel 1. Pengujian Sistem untuk Kata “Satu”

Pengucapan "satu" ke	Jarak Dengan Pola Kata					Dikenali Sebagai
	"satu"	"dua"	"tiga"	"empat"	"lima"	
1	27,8534	70,8242	65,2801	63,1374	68,0367	Satu
2	39,4622	71,8925	58,4858	58,1298	69,2063	Satu
3	31,1974	70,4965	64,9962	60,7471	69,2169	Satu
4	30,7722	72,2313	62,6032	67,2849	71,3785	Satu
5	36,1258	79,7286	75,7795	84,6157	79,1423	Satu
6	36,6424	80,204	89,583	96,3042	76,2783	Satu
7	34,4431	72,3392	80,3246	87,1547	72,6053	Satu
8	33,7968	67,1308	64,1122	64,0605	67,3958	Satu
9	32,1923	67,8813	70,5507	66,4019	64,1825	Satu
10	32,7086	73,1816	75,9395	72,7962	72,07	Satu
Akurasi						100%

Tabel 2. Pengujian Sistem untuk Kata “Dua”

Pengucapan "dua" ke	Jarak Dengan Pola Kata					Dikenali Sebagai
	"satu"	"dua"	"tiga"	"empat"	"lima"	
1	69,7265	12,9811	56,9762	50,2984	48,231	Dua
2	65,0665	17,4545	53,3988	50,2392	47,7828	Dua
3	63,7799	14,9367	56,8527	42,51	39,2812	Dua
4	60,5074	19,8398	66,8926	41,0588	40,6875	Dua
5	68,5581	31,3993	62,1716	33,1716	32,8038	Dua
6	62,9201	19,365	50,3383	44,2068	46,0204	Dua
7	61,8818	11,7066	61,9818	49,5498	41,4032	Dua
8	65,0141	12,7201	59,1851	50,8852	45,3865	Dua
9	70,7965	15,2712	50,4111	47,3372	45,824	Dua
10	61,2129	26,508	56,1364	36,2143	26,9348	Dua
Akurasi						100%

Tabel 3. Pengujian Sistem untuk Kata “Tiga”

Pengucapan "tiga" ke	Jarak Dengan Pola Kata					Dikenali Sebagai
	"satu"	"dua"	"tiga"	"empat"	"lima"	
1	65,0123	58,4501	17,9535	61,4294	50,0634	Tiga
2	66,2185	56,7854	20,0035	59,622	46,323	Tiga
3	72,5169	61,3598	11,7507	56,7890	48,0134	Tiga
4	63,086	62,4493	10,6195	55,2361	45,11	Tiga
5	70,1719	59,5151	15,7495	63,53	42,8964	Tiga
6	72,3589	58,3341	14,623	56,1170	38,4859	Tiga
7	75,9157	55,7785	20,2698	59,4136	45,5544	Tiga
8	74,0428	68,1764	13,6886	55,3514	42,7031	Tiga
9	68,7481	65,9186	21,0003	55,3453	45,7164	Tiga
10	70,5007	60,0771	15,6711	61,33	43,7008	Tiga
Akurasi						100%

Tabel 4. Pengujian Sistem untuk Kata “Empat”

Pengucapan "empat" ke	Jarak Dengan Pola Kata					Dikenali Sebagai
	"satu"	"dua"	"tiga"	"empat"	"lima"	
1	73,4448	47,4216	44,3972	26,3655	34,5127	empat
2	64,0018	46,2622	44,3113	19,9418	37,5603	empat
3	76,5542	40,0436	51,7308	23,5407	34,7205	empat
4	69,6204	44,1652	50,5418	29,0921	31,8445	empat
5	70,9657	42,3202	49,6787	25,6398	32,2213	empat
6	64,2638	47,0227	39,7574	20,2054	31,3313	empat
7	67,8388	46,7918	41,3035	19,8897	37,1892	empat
8	82,8315	48,2603	58,177	39,6166	48,3085	empat
9	67,0619	48,2297	48,8339	19,3779	35,4928	empat
10	69,4782	46,8487	48,4073	21,1669	34,4475	empat
Akurasi						100%

Tabel 5. Pengujian Sistem untuk Kata “Lima”

Pengucapan "lima" ke	Jarak Dengan Para Kata					Dikenali Sebagai
	"satu"	"dua"	"tiga"	"empat"	"lima"	
1	54,0865	44,982	49,1999	39,532	17,2156	lima
2	60,2568	48,4397	44,9351	48,7379	20,6803	lima
3	63,8611	42,1884	62,2601	40,8693	19,7958	lima
4	65,8633	53,1246	42,6219	51,8813	19,6907	lima
5	60,4459	39,3382	41,2389	40,1437	15,8345	lima
6	61,6625	45,2466	44,8056	41,8996	19,6482	lima
7	74,6767	55,6092	46,3678	55,308	18,3489	lima
8	82,3231	60,7359	43,9193	47,6013	16,43	lima
9	75,9919	47,6308	42,8351	47,7634	15,221	lima
10	82,4027	52,8856	45,6468	49,2298	15,6755	lima
Akurasi						100%

Hasil pengujian yang ditunjukkan dari tabel 1 sampai tabel 5 memperlihatkan hasil yang sempurna. Dari sepuluh kali pengucapan untuk setiap kata, sistem yang telah didisain dapat mencapai tingkat keberhasilan pengenalan dengan sempurna yaitu 100 %.

Pengujian Sistem dengan Pengucapan Kata secara Acak

Pada pengujian ini, pengucapan kata dilakukan secara acak, akan tetapi tetap setiap kata diucapkan sebanyak 10 kali sehingga total pengucapan kata sebanyak 50 kali. Jumlah kata yang diuji tetap sama yaitu lima kata dari kata “satu” sampai kata “lima”. Jadi pada prinsipnya pengujian ini sama dengan pengujian sebelumnya. Perbedaannya hanya dalam pengucapan kata. Pada pengujian sebelumnya, pengucapan kata dilakukan secara berurutan sedangkan pada pengujian ini, urutan pengucapan kata dilakukan secara acak. Berikut tabel 6 menunjukkan rangkuman hasil pengujian yang telah dilakukan.

Tabel 6. Rangkuman Hasil Pengujian Sistem dengan Pengucapan Kata secara Acak

Kata	Tingkat Keberhasilan
Satu	80 %
Dua	100 %
Tiga	90 %
Empat	90 %
Lima	100%

Dalam pengujian ini terlihat bahwa hasil yang didapat tidaklah sebagus hasil sebelumnya yang menggunakan pengucapan setiap kata secara berulang. Pada pengujian ini tingkat keberhasilan rata-rata yang dicapai oleh sistem hanya 92% saja. Hal ini mungkin disebabkan karena pada pengujian sebelumnya, pengucapan kata-kata dilakukan secara berulang sehingga konsistensi pengucapan dapat terjaga. Dengan demikian tentunya akan memberikan hasil yang lebih baik.

Pengujian Sistem untuk Pembicara yang Berbeda

Pada pengujian ini, dilakukan pengujian sistem dengan pengucapan kata-kata oleh empat orang lain yang tidak mempunyai pola referensi dalam database sistem. Komposisi empat orang tersebut terdiri dari dua wanita dan dua pria. Salah satu pria dan wanita dari empat tersebut mempunyai hubungan saudara dengan orang yang diambil suaranya sebagai database. Berikut tabel 7 sampai tabel 10 menunjukkan rangkuman hasil pengujian yang telah dilakukan untuk empat orang tersebut dengan kondisi wanita 1 dan pria 1 adalah saudara dari orang yang diambil suaranya untuk database.

Tabel 7. Rangkuman Hasil Pengujian Sistem dengan Pengucapan Kata oleh Wanita 1

Kata	Tingkat Keberhasilan
Satu	100 %
Dua	0 %
Tiga	60 %
Empat	80 %
Lima	20%

Tabel 8. Rangkuman Hasil Pengujian Sistem dengan Pengucapan Kata oleh Pria 1

Kata	Tingkat Keberhasilan
Satu	100 %
Dua	100 %
Tiga	100 %
Empat	70 %
Lima	100%

Tabel 9. Rangkuman Hasil Pengujian Sistem dengan Pengucapan Kata oleh Wanita 2

Kata	Tingkat Keberhasilan
Satu	100 %
Dua	80 %
Tiga	90 %
Empat	100 %
Lima	90%

Tabel 10. Rangkuman Hasil Pengujian Sistem dengan Pengucapan Kata oleh Pria 2

Kata	Tingkat Keberhasilan
Satu	100 %
Dua	60 %
Tiga	90 %
Empat	70 %
Lima	60%

Dari hasil pengujian yang ditunjukkan oleh tabel 7 sampai tabel 8, ternyata terlihat bahwa sistem pengenalan kata ini tidak dapat digunakan untuk mengenali suara orang lain dengan cukup baik meskipun setiap kata diucapkan 10 kali secara berulang, kecuali untuk mengenali kata dari pria1 masih memberikan hasil yang cukup baik meskipun tingkat keberhasilannya tidak mencapai 100% karena hanya terdapat tiga buah kata yang salah dari 50 kata yang diucapkan. Sedangkan sistem ini kesulitan untuk mengenali kata dari wanita1 karena hasil yang didapat hanya memiliki tingkat keberhasilan rata-rata yang kecil yaitu 52%.

Kesimpulan dan Diskusi

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem yang telah didisain dapat mengenali kata dengan cukup baik dan mencapai tingkat keberhasilan pengenalan rata-rata sebesar 92 % untuk orang yang suaranya ada dalam database. Dalam sistem ini masih terdapat kelemahan yaitu belum dapat mengenali dengan baik kata-kata yang diucapkan oleh orang lain yang tidak ada dalam database. Karena dalam disain, lama waktu sampling hanya setengah detik maka sistem ini hanya mampu mengenali kata terisolasi dengan durasi sepanjang setengah detik.

Daftar Pustaka

- [1] Rabiner, Lawrence, and Biing-Hwang Juang, 1993, "*Fundamentals of Speech Recognition*". Prentice Hall, New Jersey.
- [2] Pelton, Gordon E, 1993, "*Voice Processing*". McGraw-Hill, Inc., Singapore.
- [3] _____, 1997, "*TMS320C5x General Purpose Applications User's Guide*". Texas Instruments, Inc. Dallas.
- [4] _____, 1997, "*TMS320C5x User's Guide*". Texas Instruments, Inc., Missouri.